

A

JC512 U.S. PTO



09/16/97

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of)

ELIYAHOU HARARI, ROBERT D.)
NORMAN, and SANJAY MEHROTRA)

For: FLASH EEPROM SYSTEM)

PARENT APPLICATION
ASSIGNED TO GROUP
ART UNIT 2413

San Francisco, California

Patent Application
Hon. Commissioner of
Patents and Trademarks
Washington, D.C. 20231

CERTIFICATE OF EXPRESS MAILING UNDER 37 CFR 1.10

I hereby certify that this patent application transmittal and the documents referred to as enclosed therein are being deposited with the United States Postal Service as "Express Mail Post Office to Addressee" Mailing Label Number EM194239789US addressed to: Patent Application, Commissioner of Patents and Trademarks, Washington, D.C. 20231 on September 16, 1997.

Name: Brenda J. Dolly

Brenda J. Dolly Sept. 16, 1997
Signature Date

CONTINUATION APPLICATION TRANSMITTAL

Sir:

This is a request for filing a continuation application under 37 CFR 1.60 of pending prior application Serial No. 08/249,049, filed May 25, 1994, by ELIYAHOU HARARI, ROBERT D. NORMAN, and SANJAY MEHROTRA, for FLASH EEPROM SYSTEM.

1. Enclosed is a copy of the prior application, including the Declaration of the inventors as originally filed. I hereby certify that the attached papers are a true copy of prior application Serial No. 08/249,049 as originally filed on May 25, 1994, as appears in the files of the undersigned attorney, and that no amendments referred to in the oath or declaration (if any) filed to complete the prior application introduced new matter therein.

2. A Preliminary Amendment is enclosed.

3. The filing fee is calculated to be \$930.00, a check for which is enclosed. The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment, to Deposit Account No. 13-1030. A duplicate copy of this sheet is enclosed.

4. The prior application is assigned of record to SanDisk Corporation.

5. Five (5) sheets of formal drawings are also enclosed.

6. The Power of Attorney appearing in the prior application is to Gerald P. Parsons (Reg. No. 24,486), Martin F. Majestic (Reg. No. 25,695), J. Suzanne Siebert (Reg. No. 28,758), James S. Hsue (Reg. No. 29,545), Philip Yau (Reg. No. 32,892).


7. Address all future communications to:

Gerald P. Parsons, Esq.
MAJESTIC, PARSONS, SIEBERT & HSUE
Four Embarcadero Center, Suite 1100
San Francisco, California 94111-4106

Telephone: (415) 248-5500
Facsimile: (415) 362-5418

Dated: September 16, 1997

Docket No.: HARI.006USG


Gerald P. Parsons, Reg. No. 24,486
MAJESTIC, PARSONS, SIEBERT & HSUE
Four Embarcadero Center, Suite 1100
San Francisco, CA 94111-4106

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of)	
)	
ELIYAHOU HARARI, ROBERT D.)	Parent Application
NORMAN, and SANJAY MEHROTRA)	Assigned to Group Art
)	Unit: 2413
Serial No.: UNASSIGNED)	
)	
Filed: HEREWITH)	
)	
For: FLASH EEPROM SYSTEM)	
)	San Francisco, California

Hon. Commissioner of
Patents and Trademarks
Washington, D.C. 20231

PRELIMINARY AMENDMENT

Sir:

Please preliminarily amend the above-identified new continuation application, as follows:

IN THE SPECIFICATION:

Page 1, before line 4, add the following section:

--Cross-Reference to Related Applications

This is a continuation of application Serial No. 08/249,049, filed May 25, 1994, now patent no. 5,671,229, which is a continuation of application Serial No. 07/963,837, filed October 20, 1992, now abandoned, which in turn is a division of application serial no 07/337,566, filed April 13, 1989, now abandoned.--

IN THE CLAIMS:

Cancel claims 1-62, without prejudice, and substitute the following new claims 63-67 therefore:

08/249,049

0893433 094697

--63. A method for extending the life of Flash EEprom memory in a Flash EEprom system, comprising the steps of:

temporarily storing data files from a host system intended for the Flash EEprom memory in a cache memory, said cache memory able to undergo significantly more write/erase cycles than the Flash EEprom memory;

writing data files into the cache memory instead of the Flash EEprom memory in response to a write request from the host system;

determining the time since each data file was last written into said cache memory; and

moving first a data file having the longest time since last written first from the cache memory to the Flash EEprom memory when additional space for new data files is required in the cache memory, thereby substantially reducing the number of actual writes and associated stress to the Flash EEprom memory.

64. A method for extending the life of Flash EEprom memory in a Flash EEprom system, comprising the steps of:

temporarily storing data files from a host system intended for the Flash EEprom memory in a cache memory, said cache memory able to undergo significantly more write/erase cycles than the Flash EEprom memory;

writing data files into the cache memory instead of the Flash EEprom memory in response to a write request from the host system;

storing the identity of data files and the time each data file was last written into said cache memory in a tag memory; and

by reference to the tag memory, moving data file having the longest time since last written first from the cache memory to the Flash EEPROM memory when additional space for new data files is required in the cache memory, thereby substantially reducing the number of actual writes and associated stress to the Flash EEPROM memory.

65. A method for extending the life of Flash EEPROM memory in a Flash EEPROM system, comprising the steps of:

temporarily storing data files from a host system intended for the Flash EEPROM memory in a cache memory, said cache memory able to undergo significantly more write/erase cycles than the Flash EEPROM memory;

in response to a write request from the host system, writing a data file either into the Flash EEPROM memory when a previous copy of said data file is not present in the cache memory, or into the cache memory when a previous copy of said data file is present in the cache memory; and

moving data file having the longest time since last written first from the cache memory to the Flash EEPROM memory when additional space for new data files is required in the cache memory, thereby substantially reducing the number of actual writes and associated stress to the Flash EEPROM memory.

66. A method for extending the life of Flash EEPROM memory in a Flash EEPROM system, comprising the steps of:

temporarily storing data files from a host system intended for the Flash EEPROM memory in a cache memory, said cache memory able to undergo significantly more write/erase cycles than the Flash EEPROM memory;

in response to a write request from the host system, writing a data file either into the Flash EEPROM memory when said data file is last written after a predetermined period of time, or into the cache memory when said data file is last written within the predetermined period of time; and

moving data file having the longest time since last written first from the cache memory to the Flash EEPROM memory when additional space for new data files is required in the cache memory, thereby substantially reducing the number of actual writes and associated stress to the Flash EEPROM memory.

67. A method for extending the life of Flash EEPROM memory in a Flash EEPROM system, comprising the steps of:

temporarily storing data files from a host system intended for the Flash EEPROM memory in a cache memory, said cache memory able to undergo significantly more write/erase cycles than the Flash EEPROM memory;

storing the identity of data files and the time each data file was last written into said cache memory in a tag memory;

in response to a write request from the host system, writing a data file into the Flash EEPROM memory when the data file is not identified in the tag memory, or into the cache memory when the data file is identified in the tag memory; and

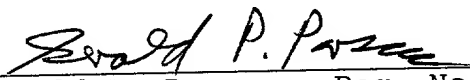
by reference to the tag memory, moving data file having the longest time since last written first from the cache memory to the Flash EEPROM memory when additional space for new data files is required in the cache memory, thereby substantially reducing the number of actual writes and associated stress to the Flash EEPROM memory.--

REMARKS

This continuation application is being filed to pursue claims to the use of a write cache memory in a flash EEprom (Electrically Erasable and Programmable Read-Only-Memory) system. Claims 63-67 being substituted for the original application claims are the same as those withdrawn from consideration in related application Serial No. 08/469,873, filed June 6, 1995, now abandoned. This write cache memory subject matter was also prosecuted in parent application Serial No. 07/337,566, filed April 13, 1989, now abandoned.

A prompt and favorable examination of this application is solicited.

Dated: September 16, 1997 Respectfully submitted,


Gerald P. Parsons, Reg. No. 24,486
MAJESTIC, PARSONS, SIEBERT & HSUE
Four Embarcadero Center, Suite 1100
San Francisco, CA 94111-4106
Telephone: (415) 248-5500
Facsimile: (415) 362-5418

Atty. Docket: HARI.006USG

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
APPLICATION FOR PATENT

Title: FLASH EEprom SYSTEM

Inventors: Eliyahou Harari, Robert D. Norman,
Sanjay Mehrotra

Background of the Invention

5 This invention relates generally to semiconductor electrically erasable programmable read only memories (EEprom), and specifically to a system of integrated circuit Flash EEprom chips.

10 Computer systems typically use magnetic disk drives for mass storage of data. However, disk drives are disadvantageous in that they are bulky and in their requirement for high precision moving mechanical parts. Consequently they are not rugged and are prone to reliability problems, as well as consuming significant
15 amounts of power. Solid state memory devices such as DRAM's and SRAM's do not suffer from these disadvantages. However, they are much more expensive, and require constant power to maintain their memory (volatile). Consequently, they are typically used as
20 temporary storage.

 EEprom's and Flash EEprom's are also solid state memory devices. Moreover, they are nonvolatile, and retain their memory even after power is shut down. However, conventional Flash EEprom's have a limited
25 lifetime in terms of the number of write (or program)/erase cycles they can endure. Typically the devices are rendered unreliable after 10^2 to 10^3 write/erase cycles. Traditionally, they are typically used in applications where semi-permanent storage of
30 data or program is required but with a limited need for reprogramming.

Accordingly, it is an object of the present invention to provide a Flash EEPROM memory system with enhanced performance and which remains reliable after enduring a large number of write/erase cycles.

5 It is another object of the present invention to provide an improved Flash EEPROM system which can serve as non-volatile memory in a computer system.

10 It is another object of the present invention to provide an improved Flash EEPROM system that can replace magnetic disk storage devices in computer systems.

It is another object of the present invention to provide a Flash EEPROM system with improved erase operation.

15 It is another object of the present invention to provide a Flash EEPROM system with improved error correction.

20 It is yet another object of the present invention to provide a Flash EEPROM with improved write operation that minimizes stress to the Flash EEPROM device.

It is still another object of the present invention to provide a Flash EEPROM system with enhanced write operation.

25 Summary of the Invention

These and additional objects are accomplished by improvements in the architecture of a system of EEPROM chips, and the circuits and techniques therein.

30 According to one aspect of the present invention, an array of Flash EEPROM cells on a chip is organized into sectors such that all cells within each sector are erasable at once. A Flash EEPROM memory system comprises one or more Flash EEPROM chips under the control of a controller. The invention allows any

0894133-094697

defective cells in a sector exceeds a predetermined number, the sector containing the defective cells is replaced by a substitute sector.

5 An important feature of the invention allows defective cells or defective sectors to be remapped as soon as they are detected thereby enabling error correction codes to adequately rectify the relatively few errors that may crop up in the system.

10 According to yet another aspect of the present invention, a write cache is used to minimize the number of writes to the Flash EEPROM memory. In this way the Flash EEPROM memory will be subject to fewer stress inducing write/erase cycles, thereby retarding its aging. The most active data files are written to the
15 cache memory instead of the Flash EEPROM memory. Only when the activity levels have reduced to a predetermined level are the data files written from the cache memory to the Flash EEPROM memory. Another advantage of the invention is the increase in write throughput by virtue
20 of the faster cache memory.

According to yet another aspect of the present invention, one or more printed circuit cards are provided which contain controller and EEPROM circuit chips for use in a computer system memory for long term,
25 non-volatile storage, in place of a hard disk system, and which incorporate various of the other aspects of this invention alone and in combination.

Additional objects, features, and advantages of the present invention will be understood from the
30 following description of its preferred embodiments, which description should be taken in conjunction with the accompanying drawings.

Brief Description of the Drawings

Fig. 1A is a general microprocessor system including the Flash EEprom memory system of the present invention:

5 Fig. 1B is schematic block diagram illustrating
a system including a number of Flash EEprom memory chips
and a controller chip;

Fig. 2 is a schematic illustration of a system of Flash EEprom chips, among which memory sectors are selected to be erased;

Fig. 3A is a block circuit diagram (in the controller) for implementing selective multiple sector erase according to the preferred embodiment;

Fig. 3B shows details of a typical register
15 used to select a sector for erase as shown in Fig. 2A;

Fig. 4 is a flow diagram illustrating the erase sequence of selective multiple sector erase;

Fig. 5 is a schematic illustration showing the partitioning of a Flash EEprom sector into a data area and a spare redundant area;

Fig. 6 is a circuit block diagram illustrating the data path control during read operation using the defect mapping scheme of the preferred embodiment;

Fig. 7 is a circuit block diagram illustrating the data path control during the write operation using the defect mapping scheme of the preferred embodiment;

Fig. 8 is a block diagram illustrating the write cache circuit inside the controller.

Description of the Preferred EmbodimentsEEprom System

5 A computer system in which the various aspects of the present invention are incorporated is illustrated generally in Figure 1A. A typical computer system architecture includes a microprocessor 21 connected to a system bus 23, along with random access, main system memory 25, and at least one or more input-output devices 27, such as a keyboard, monitor, modem, and the like. 10 Another main computer system component that is connected to a typical computer system bus 23 is a large amount of long-term, non-volatile memory 29. Typically, such a memory is a disk drive with a capacity of tens of megabytes of data storage. This data is retrieved into 15 the system volatile memory 25 for use in current processing, and can be easily supplemented, changed or altered.

One aspect of the present invention is the substitution of a specific type of semiconductor memory system for the disk drive but without having to sacrifice non-volatility, ease of erasing and rewriting data into the memory, speed of access, low cost and reliability. This is accomplished by employing an array of electrically erasable programmable read only memories (EEprom's) integrated circuit chips. This type of 25 memory has additional advantages of requiring less power to operate, and of being lighter in weight than a hard disk drive magnetic media memory, thereby being especially suited for battery operated portable computers. 30

The bulk storage memory 29 is constructed of a memory controller 31, connected to the computer system bus 23, and an array 33 of EEprom integrated circuit chips. Data and instructions are communicated from the 35 controller 31 to the EEprom array 33 primarily over a

2025 RELEASE UNDER E.O. 14176

serial data line 35. Similarly, data and status signals are communicated from the EEprom 33 to the controller 31 over serial data lines 37. Other control and status circuits between the controller 31 and the EEprom array 33 are not shown in Figure 1A.

Referring to Figure 1B, the controller 31 is preferably formed primarily on a single integrated circuit chip. It is connected to the system address and data bus 39, part of the system bus 33, as well as being connected to system control lines 41, which include interrupt, read, write and other usual computer system control lines.

The EEprom array 33 includes a number of EEprom integrated circuit chips 43, 45, 47, etc. Each includes a respective chip select and enable line 49, 51 and 53 from interface circuits 40. The interface circuits 40 also act to interface between the serial data lines 35, 37 and a circuit 55. Memory location addresses and data being written into or read from the EEprom chips 43, 45, 47, etc. are communicated from a bus 55, through logic and register circuits 57 and thence by another bus 59 to each of the memory chips 43, 45, 47 etc.

The bulk storage memory 29 of Figures 1A and 1B can be implemented on a single printed circuit card for moderate memory sizes. The various lines of the system buses 39 and 41 of Figure 1B are terminated in connecting pins of such a card for connection with the rest of the computer system through a connector. Also connected to the card and its components are various standard power supply voltages (not shown).

For large amounts of memory, that which is conveniently provided by a single array 33 may not be enough. In such a case, additional EEprom arrays can be connected to the serial data lines 35 and 37 of the controller chip 31, as indicated in Figure 1B. This is

preferably all done on a single printed circuit card but if space is not sufficient to do this, then one or more EEprom arrays may be implemented on a second printed circuit card that is physically mounted onto the first and connected to a common controller chip 31.

Erase of Memory Structures

In system designs that store data in files or blocks the data will need to be periodically updated with revised or new information. It may also be desirable to overwrite some no longer needed information, in order to accommodate additional information. In a Flash EEprom memory, the memory cells must first be erased before information is placed in them. That is, a write (or program) operation is always preceded by an erase operation.

In conventional Flash erase memory devices, the erase operation is done in one of several ways. For example, in some devices such as the Intel corporation's model 27F-256 CMOS Flash EEprom, the entire chip is erased at one time. If not all the information in the chip is to be erased, the information must first be temporarily saved, and is usually written into another memory (typically RAM). The information is then restored into the nonvolatile Flash erase memory by programming back into the device. This is very slow and requires extra memory as holding space.

In other devices such as Seeq Technology Incorporated's model 48512 Flash EEprom chip, the memory is divided into blocks (or sectors) that are each separately erasable, but only one at a time. By selecting the desired sector and going through the erase sequence the designated area is erased. While, the need for temporary memory is reduced, erase in various areas of the memory still requires a time consuming sequential

approach.

In the present invention, the Flash EEPROM memory is divided into sectors where all cells within each sector are erasable together. Each sector can be addressed separately and selected for erase. One important feature is the ability to select any combination of sectors for erase together. This will allow for a much faster system erase than by doing each one independently as in prior art.

Figure 2 illustrates schematically selected multiple sectors for erase. A Flash EEPROM system includes one or more Flash EEPROM chips such as 201, 203, 205. They are in communication with a controller 31 through lines 209. Typically, the controller 31 is itself in communication with a microprocessor system (not shown). The memory in each Flash EEPROM chip is partitioned into sectors where all memory cells within a sector are erasable together. For example, each sector may have 512 byte (i.e. 512x8 cells) available to the user, and a chip may have 1024 sectors. Each sector is individually addressable, and may be selected, such as sectors 211, 213, 215, 217 in a multiple sector erase. As illustrated in figure 2, the selected sectors may be confined to one EEPROM chip or be distributed among several chips in a system. The sectors that were selected will all be erased together. This capability will allow the memory and system of the present invention to operate much faster than the prior art architectures.

Figure 3A illustrates a block diagram circuit 220 on a Flash EEPROM chip (such as the chip 201 of figure 2) with which one or more sectors such as 211, 213 are selected (or deselected) for erase. Essentially, each sector such as 211, 213 is selected or tagged by setting the state of an erase enable register

such as 221, 223 associated with the respective sectors. The selection and subsequent erase operations are performed under the control of the controller 31 (see figure 2). The circuit 220 is in communication with the controller 31 through lines 209. Command information from the controller is captured in the circuit 220 by a command register 225 through a serial interface 227. It is then decoded by a command decoder 229 which outputs various control signals. Similarly, address information is captured by an address register 231 and is decoded by an address decoder 233.

For example, in order to select the sector 211 for erase, the controller sends the address of the sector 211 to the circuit 220. The address is decoded in line 235 and is used in combination with a set erase enable signal in bus 237 to set an output 239 of the register 221 to HIGH. This enables the sector 211 in a subsequent erase operation. Similarly, if the sector 213 is also desired to be erased, its associated register 223 may be set HIGH.

Figure 3B shows the structure of the register such as 221, 223 in more detail. The erase enable register 221 is a SET/RESET latch. Its set input 241 is obtained from the set erase enable signal in bus 237 gated by the address decode in line 235. Similarly, the reset input 243 is obtained from the clear erase enable signal in bus 237 gated by the address decode in line 235. In this way, when the set erase enable signal or the clear erase enable signal is issued to all the sectors, the signal is effective only on the sector that is being addressed.

After all sectors intended for erase have been selected, the controller then issues to the circuit 220, as well as all other chips in the system a global erase command in line 251 along with the high voltage for

erasing in line 209. The device will then erase all the sectors that have been selected (i.e. the sectors 211 and 213) at one time. In addition to erasing the desired sectors within a chip, the architecture of the present system permits selection of sectors across various chips for simultaneous erase.

Figures 4(1)-4(11) illustrate the algorithm used in conjunction with the circuit 220 of figure 3A. In figure 4(1), the controller will shift the address into the circuit 220 which is decoded in the line to the erase enable register associated with the sector that is to be erased. In figure 4(2), the controller shifts in a command that is decoded to a set erase enable command which is used to latch the address decode signal onto the erase enable register for the addressed sector. This tags the sector for subsequent erase. In figure 4(3), if more sectors are to be tagged, the operations described relative to figures 4(1)-4(2) are repeated until all sectors intended for erase have been tagged. After all sectors intended for erase have been tagged, the controller initiates an erase cycle as illustrated in figure 4(4).

Optimized erase implementations have been disclosed in two copending U.S. patent applications. They are copending U.S. patent applications, Serial No. 204,175, filed June 8, 1988, by Dr. Eliyahou Harari and one entitled "Multi-State EEprom Read and Write Circuits and Techniques," filed on the same day as the present application, by Sanjay Mehrotra and Dr. Eliyahou Harari. The disclosures of the two applications are hereby incorporate by reference. The Flash EEprom cells are erased by applying a pulse of erasing voltage followed by a read to verify if the cells are erased to the "erased" state. If not, further pulsing and verifying

are repeated until the cells are verified to be erased. By erasing in this controlled manner, the cells are not subject to over-erasure which tends to age the EEprom device prematurely as well as make the cells harder to program.

As the group of selected sectors is going through the erase cycle, some sectors will reach the "erase" state earlier than others. Another important feature of the present invention is the ability to remove those sectors that have been verified to be erased from the group of selected sectors, thereby preventing them from over-erasing.

Returning to figure 4(4), after all sectors intended for erase have been tagged, the controller initiates an erase cycle to erase the group of tagged sectors. In figure 4(5), the controller shifts in a global command called Enable Erase into each Flash EEprom chip that is to perform an erase. This is followed in figure 4(5) by the controller raising of the erase voltage line (V_e) to a specified value for a specified duration. The controller will lower this voltage at the end of the erase duration time. In figure 4(6), the controller will then do a read verify sequence on the sectors selected for erase. In figure 4(7), if none of the sectors are verified, the sequences illustrated in figures 4(5)-4(7) are repeated. In figures 4(8) and 4(9), if one or more sectors are verified to be erased, they are taken out of the sequence. Referring also to figure 3A, this is achieved by having the controller address each of the verified sectors and clear the associated erase enable registers back to a LOW with a clear enable command in bus 237. The sequences illustrated in figures 4(5)-4(10) are repeated until all the sectors in the group are verified to be erased in figure 4(11). At the completion of the

erase cycle, the controller will shift in a No Operation (NOP) command and the global Enable Erase command will be withdrawn as a protection against a false erasure.

The ability to select which sectors to erase and which ones not to, as well as which ones to stop erasing is advantageous. It will allow sectors that have erased before the slower erased sectors to be removed from the erase sequence so no further stress on the device will occur. This will increase the reliability of the system. Additional advantage is that if a sector is bad or is not used for some reason, that sector can be skipped over with no erase occurring within that sector. For example, if a sector is defective and have shorts in it, it may consume much power. A significant system advantage is gained by the present invention which allows it to be skipped on erase cycles so that it may greatly reduce the power required to erase the chip.

Another consideration in having the ability to pick the sectors to be erased within a device is the power savings to the system. The flexibility in erase configuration of the present invention enables the adaptation of the erase needs to the power capability of the system. This can be done by configuring the systems to be erased differently by software on a fixed basis between different systems. It also will allow the controller to adaptively change the amount of erasing being done by monitoring the voltage level in a system, such as a laptop computer.

30 An additional performance capability of the
system in the present invention is the ability to issue
a reset command to a Flash EEprom chip which will clear
all erase enable latches and will prevent any further
erase cycles from occurring. This is illustrated in
35 figures 2A and 2B by the reset signal in the line 261.

By doing this in a global way to all the chips, less time will be taken to reset all the erase enable registers.

5 An additional performance capability is to have the ability to do erase operations without regard to chip select. Once an erase is started in some of the memory chips, the controller in the system can access other memory chips and do read and write operations on them. In addition, the device(s) doing the erase can be
10 selected and have an address loaded for the next command following the erase.

Defect Mapping

Physical defects in memory devices give rise to hard errors. Data becomes corrupted whenever it is
15 stored in the defective cells. In conventional memory devices such as RAM's and Disks, any physical defects arising from the manufacturing process are corrected at the factory. In RAM's, spare redundant memory cells on chip may be patched on, in place of the defective cells.
20 In the traditional disk drive, the medium is imperfect and susceptible to defects. To overcome this problem manufacturers have devised various methods of operating with these defects present, the most usual being defect mapping of sectors. In a normal disk system the media
25 is divided into cylinders and sectors. The sector being the basic unit in which data is stored. When a system is partitioned into the various sectors the sectors containing the defects are identified and are marked as bad and not to be used by the system. This is done in
30 several ways. A defect map table is stored on a particular portion of the disk to be used by the interfacing controller. In addition, the bad sectors are marked as bad by special ID and flag markers. When the defect is addressed, the data that would normally be

stored there is placed in an alternative location. The requirement for alternative sectors makes the system assign spare sectors at some specific interval or location. This reduces the amount of memory capacity
5 and is a performance issue in how the alternative sectors are located.

One important application of the present invention is to replace a conventional disk storage device with a system incorporating an array of Flash
10 EEprom memory chips. The EEprom system is preferably set up to emulate a conventional disk, and may be regarded as a "solid-state disk".

In a "disk" system made from such solid-state memory devices, low cost considerations necessitate
15 efficient handling of defects. Another important feature of the invention enables the error correction scheme to conserve as much memory as possible. Essentially, it calls for the defective cells to be remapped cell by cell rather than by throwing away the
20 whole sector (512 bytes typically) whenever a defect occurs in it. This scheme is especially suited to the Flash EEprom medium since the majority of errors will be bit errors rather than a long stream of adjacent defects as is typical in traditional disk medium.

25 In both cases of the prior art RAM and magnetic disk, once the device is shipped from the factory, there is little or no provision for replacing hard errors resulting from physical defects that appear later during normal operation. Error corrections then mainly rely on
30 schemes using error correction codes (ECC).

The nature of the Flash EEprom device predicates a higher rate of cell failure especially with increasing program/erase cycling. The hard errors that accumulate with use would eventually overwhelm the ECC
35 and render the device unusable. One important feature

009113 09169

of the present invention is the ability for the system to correct for hard errors whenever they occur. Defective cells are detected by their failure to program or erase correctly. Also during read operation, defective cells are detected and located by the ECC. As soon as a defective cell is identified, the controller will apply defect mapping to replace the defective cell with a space cell located usually within the same sector. This dynamic correction of hard errors, in addition to conventional error correction schemes, significantly prolongs the life of the device.

Another feature of the present invention is an adaptive approach to error correction. Error correction code (ECC) is employed at all times to correct for soft errors as well as any hard errors that may arise. As soon as a hard error is detected, defect mapping is used to replace the defective cell with a spare cell in the same sector block. Only when the number of defective cells in a sector exceeds the defect mapping's capacity for that specific sector will the whole sector be replaced as in a conventional disk system. This scheme minimized wastage without compromising reliability.

Figure 5 illustrates the memory architecture for the cell remapping scheme. As described before, the Flash EEPROM memory is organized into sectors where the cells in each sector are erasable together. The memory architecture has a typical sector 401 organized into a data portion 403 and a spare (or shadow) portion 405. The data portion 403 is memory space available to the user. The spare portion 405 is further organized into an alternative defects data area 407, a defect map area 409, a header area 411 and an ECC and others area 413. These areas contain information that could be used by the controller to handle the defects and other overhead information such as headers and ECC.

Whenever a defective cell is detected in the sector, a good cell in the alternative defects data area 407 is assigned to backup the data designated for the defective cell. Thus even if the defective cell stores the data incorrectly, an error-free copy is stored in the backup cell. The addresses of the defective cell and the backup cell are stored as defect pointers in the defect map 409.

It is to be understood that the partitioning between the user data portion 403 and the spare portion 405 need not be rigid. The relative size of the various partitioned areas may be logically reassigned. Also the grouping of the various areas is largely for the purpose of discussion and not necessarily physically so. For example, the alternative defects data area 407 has been schematically grouped under the spare portion 405 to express the point that the space it occupies is no longer available to the user.

In a read operation, the controller first reads the header, the defect map and the alternative defects data. It then reads the actual data. It keeps track of defective cells and the location of the substitute data by means of the defect map. Whenever a defective cell is encountered, the controller substitutes its bad data with the good data from the alternative defects.

Figure 6 illustrates the read data path control in the preferred embodiment. A memory device 33 which may include a plurality of Flash EEPROM chips is under the control of the controller 31. The controller 31 is itself part of a microcomputer system under the control of a microprocessor (not shown). To initiate the reading of a sector, the microprocessor loads a memory address generator 503 in the controller with a memory address for starting the read operation. This information is loaded through a microprocessor interface

08931133 091697

5

10

30

35

as the data bits are read.

Once the Header has been determined to be a match and the defect pointers and alternative bits have been loaded, the controller begins to shift out the address of the lowest address of the desired sector to be read. The data from the sector in the memory device 33 is then shifted into the controller chip 31. The receiver 515 converts the data to a parallel format and transfers each byte into a temporary holding FIFO 519 to be shipped out of the controller.

A pipeline architecture is employed to provide efficient throughput as the data is gated through the controller from the receiver 515 to the FIFO 519. As each data bit is received from memory the controller is comparing the address of the data being sent (stored in the address generator 507) against the defect pointer map (stored in the register file 509). If the address is determined to be a bad location, by a match at the output of the comparator 521, the bad bit from the memory received by the receiver 515 is replaced by the good bit for that location. The good bit is obtained from the alternative defects data file 517. This is done by switching the multiplexer 523 to receive the good bit from the alternative defects data file instead of the bad bit from the receiver 515, as the data is sent to the FIFO 519. Once the corrected data is in the FIFO it is ready to be sent to buffer memory or system memory (not shown). The data is sent from the controller's FIFO 519 to the system memory by the controller's DMA controller 507. This controller 507 then requests and gets access to the system bus and puts out an address and gates the data via an output interface 525 out to the system bus. This is done as each byte gets loaded into the FIFO 519. As the corrected data is loaded into the FIFO it will also be

00000000000000000000000000000000

gated into an ECC hardware 527 where the data file will be acted on by the ECC.

Thus in the manner described, the data read from the memory device 33 is gated through the controller 31 to be sent to the system. This process continues until the last bit of addressed data has been transferred.

In spite of defect mapping of previously detected defective cells, new hard errors might occur since the last mapping. As the dynamic defect mapping constantly "puts away" new defective cells, the latest hard error that may arise between defect mapping would be adequately handled by the ECC. As the data is gated through the controller 31, the controller is gating the ECC bits into the ECC hardware 527 to determine if the stored value matched the just calculated remainder value. If it matches then the data transferred out to the system memory was good and the read operation was completed. However, if the ECC registers an error then a correction calculation on the data sent to system memory is performed and the corrected data re-transmitted. The method for calculating the error can be done in hardware or software by conventional methods. The ECC is also able to calculate and locate the defective cell causing the error. This may be used by the controller 31 to update the defect map associated with the sector in which the defective cell is detected. In this manner, hard errors are constantly removed from the Flash EEprom system.

Figure 7 illustrates the write data path control in the preferred embodiment. The first portion of a write sequence is similar to a read sequence described previously. The microprocessor first loads the Address pointers for the memory device 33 and the DMA as in the read sequence. It also loads the header

desired into the address generator 503 and the command queue into the command sequencer 511. The command queue is loaded with a read header command first. Thereafter, control is passed over to the controller 31. The controller then gates the address and command to the memory device 33, as in the read sequence. The memory device returns header data through controller's receiver 515. The controller compares the received header data to the expected value (stored in the holding register file 509). If the compare is correct, the proper location is verified and the sequence continues. Then the controller loads the defective address pointers from the memory device 33 into the holding register file 509 and the alternative data into the alternative defects data file 517.

Next, the controller begins to fetch the write data from system memory (not shown). It does this by getting access to the system bus, outputs the memory or bus address and does the read cycle. It pulls the data into a FIFO 601 through an input interface 603. The controller then shifts the starting sector address (lowest byte address) from the address generator 503 to the selected memory device 33. This is followed by data from the FIFO 601. These data are routed through multiplexers 605 and 513 and converted to serial format before being sent to the memory device 33. This sequence continues until all bytes for a write cycle have been loaded into the selected memory.

A pipeline architecture is employed to provide efficient throughput as the data is gated from the FIFO 601 to the selected memory 33. The data gated out of the FIFO 601 is sent to the ECC hardware 527 where a remainder value will be calculated within the ECC. In the next stage, as the data is being sent to the memory device through multiplexers 605 and 513, the comparator

521 is comparing its address from the address generator 503 to the defect pointer address values in the holding register file 509. When a match occurs, indicating that a defective location is about to be written, the controller saves this bit into the alternative defect data file 517. At the same time, all bad bits sent to memory will be sent as zeroes.

After the bytes for a write cycle have been loaded into the selected memory device, the controller issues a program command to the memory device and initiate a write cycle. Optimized implementations of write operation for Flash EEPROM device have been disclosed in two previously cited co-pending U.S. patent applications, Serial No. 204,175, and one entitled "Multi-State EEPROM Read and Write Circuits and Techniques." Relevant portions of the disclosures are hereby incorporated by reference. Briefly, during the write cycle, the controller applies a pulse of programming (or writing) voltages. This is followed by a verify read to determine if all the bits have been programmed properly. If the bits did not verify, the controller repeats the program/verify cycle until all bits are correctly programmed.

If a bit fails to verify after prolonged program/verify cycling, the controller will designate that bit as defective and update the defect map accordingly. The updating is done dynamically, as soon as the defective cell is detected. Similar actions are taken in the case of failure in erase verify.

After all the bits have been programmed and verified, the controller loads the next data bits from the FIFO 601 and addresses the next location in the addressed sector. It then performs another program/verify sequence on the next set of bytes. The sequence continues until the end of the data for that

sector. Once this has occurred, the controller addresses the shadow memory (header area) associated with the sector (see figure 5) and writes the contents of the ECC registers into this area.

5 In addition, the collection of bits that was
flagged as defective and were saved in the alternative
defects data file 516 is then written in memory at the
alternative defects data locations (see figure 5),
thereby saving the good bit values to be used on a
10 subsequent read. Once these data groups are written and
verified, the sector write is considered completed.

The present invention also has provision for defect mapping of the whole sector, but only after the number of defective cells in the sector has exceeded the cell defect mapping's capacity for that specific sector. A count is kept of the number of defective cells in each sector. When the number in a sector exceeds a predetermined value, the controller marks that sector as defective and maps it to another sector. The defect pointer for the linked sectors may be stored in a sector defect map. The sector defect map may be located in the original defective sector if its spare area is sufficiently defect-free. However, when the data area of the sector has accumulated a large number of defects, it is quite likely that the spare area will also be full of defects.

Thus, it is preferable in another embodiment to locate the sector map in another memory maintained by the controller. The memory may be located in the controller hardware or be part of the Flash EEprom memory. When the controller is given an address to access data, the controller compares this address against the sector defect map. If a match occurs then access to the defective sector is denied and the substitute address present in the defect map is entered,

and the corresponding substitute sector is accessed instead.

In yet another embodiment, the sector remapping is performed by the microprocessor. The microprocessor
5 looks at the incoming address and compares it against the sector defect map. If a match occurs, it does not issue the command to the controller but instead substitute the alternative location as the new command.

Apart from the much higher speed of the solid-
10 state disk, another advantage is the lack of mechanical parts. The long seek times, rotational latency inherent in disk drives are not present. In addition, the long synchronization times, sync mark detects and write gaps are not required. Thus the overhead needed for
15 accessing the location where data is to be read or written is much less. All of these simplifications and lack of constraints result in a much faster system with much reduced overheads. In addition, the files can be arranged in memory in any address order desired, only
20 requiring the controller to know how to get at the data as needed.

Another feature of the invention is that defect mapping is implemented without the need to interrupt the data stream transferred to or from the sector. The data
25 in a block which may contain errors are transferred regardless, and is corrected afterwards. Preserving the sequential addressing will result in higher speed by itself. Further, it allows the implementation of an efficient pipeline architecture in the read and write
30 data paths.

Write Cache System

Cache memory is generally used to speed up the performance of systems having slower access devices. For example in a computer system, access of data from

0893443 091597

by excessive program/erase cycles. Even with the improved Flash EEPROM memory device as disclosed in co-pending U.S. patent applications, Serial No. 204,175 and one entitled "Multi-state EEPROM Read and Write Circuits and Techniques," by Sanjay Mehrotra and Dr. Eliyahou Harari filed on the same day as the present application, the endurance limit is approximately 10^6 program/erase cycles. In a ten-year projected life time of the device, this translates to a limit of one program/erase cycle per 5 minutes. This may be marginal in normal computer usage.

To overcome this problem, a cache memory is used in a novel way to insulate the Flash EEPROM memory device from enduring too many program/erase cycles. The primary function of the cache is to act on writes to the Flash EEPROM memory and not on reads of the Flash EEPROM memory, unlike the case with traditional caches. Instead of writing to the Flash EEPROM memory every time the data is updated, the data may be operated on several times in the cache before being committed to the Flash EEPROM memory. This reduces the number of writes to the Flash EEPROM memory. Also, by writing mostly into the faster cache memory and reducing the number of writes to the slower Flash EEPROM, an additional benefit is the increase in system write throughput.

A relatively small size cache memory is quite effective to implement the present invention. This helps to overcome the problem of data loss in the volatile cache memory during a power loss. In that event, it is relatively easy to have sufficient power reserve to maintain the cache memory long enough and have the data dumped into a non-volatile memory such as a specially reserved space in the Flash EEPROM memory. In the event of a power down or and power loss to the system, the write cache system may be isolated from the

system and a dedicated rechargeable power supply may be switch in only to power the cache system and the reserved space in the Flash EEprom memory.

Figure 8 illustrates schematically a cache system 701 as part of the controller, according to the present invention. On one hand the cache system 701 is connected to the Flash EEprom memory array 33. On the other hand it is connected to the microprocessor system (not shown) through a host interface 703. The cache system 701 has two memories. One is a cache memory 705 for temporarily holding write data files. The other is a tag memory 709 for storing relevant information about the data files held in the cache memory 705. A memory timing/control circuit 713 controls the writing of data files from the cache memory 705 to the Flash EEprom memory 33. The memory control circuit 713 is responsive to the information stored in the tag memory as well as a power sensing input 715 which is connected through the host interface 703 via a line 717 to the power supply of the microprocessor system. A power loss in the microprocessor system will be sensed by the memory control circuit 713 which will then download all the data files in the volatile cache memory 705 to the non-volatile Flash EEprom memory 33.

In the present invention, the Flash EEprom memory array 33 is organized into sectors (typically 512 byte size) such that all memory cells within each sector are erasable together. Thus each sector may be considered to store a data file and a write operation on the memory array acts on one or more such files.

During read of a new sector in the Flash EEprom memory 33, the data file is read out and sent directly to the host through the controller. This file is not used to fill the cache memory 705 as is done in the traditional cache systems.

After the host system has processed the data within a file and wishes to write it back to the Flash EEprom memory 33, it accesses the cache system 701 with a write cycle request. The controller then intercepts
5 this request and acts on the cycle.

In one embodiment of the invention, the data file is written to the cache memory 705. At the same time, two other pieces of information about the data file are written to a tag memory 709. The first is a
10 file pointer which identifies the file present in the cache memory 705. The second is a time stamp that tells what time the file was last written into the cache memory. In this way, each time the host wishes to write to the Flash EEprom memory 33, the data file is actually
15 first stored in the cache memory 705 along with pointers and time stamps in the tag memory 709.

In another embodiment of the invention, when a write from the host occurs, the controller first checks to see if that file already existed in the cache memory
20 705 or has been tagged in the tag memory 709. If it has not been tagged, the file is written to the Flash memory 33, while its identifier and time stamp are written to the tag memory 709. If the file already is present in the cache memory or has been tagged, it is updated in
25 the cache memory and not written to the Flash memory. In this way only infrequently used data files are written into the Flash memory while frequently used data files are trapped in the cache memory.

In yet another embodiment of the invention,
30 when a write from the host occurs, the controller first checks to see if that data file has been last written anywhere within a predetermined period of time (for example, 5 minutes). If it has not, the data file is written to the Flash memory 33, while its identifier
35 and time stamp are written to the tag memory 709. If

2025 RELEASE UNDER E.O. 14176

5

10

20

25

35

Two things can happen to a file's time stamp, depending on the activity of the file. One possibility is for the time stamp to be reset in the event of a new activity occurring. The other possibility is that no
5 new activity occurs for the file and the time stamp continues to increment until the file is removed from the cache. In practice a maximum limit may be reached if the time stamp is allowed to increase indefinitely. For example, the system may allow the time stamp to
10 increment to a maximum period of inactivity of 5 minutes. Thus, when a data file is written in the cache memory, the time stamp for the file is set at its initial value. Then the time stamp will start to age, incrementing at every time step unless reset to its
15 initial value again by another write update. After say, 5 minutes of inactivity, the time stamp has incremented to a maximum terminal count.

In one embodiment of keeping count, a bit can be shifted one place in a shift register each time a
20 count increment for a file occurs. If the file is updated (a new activity has occurred) the bit's location will be reset to the initial location of the shift register. On the other hand, if the file remains inactive the bit will eventually be shifted to the
25 terminal shift position. In another embodiment, a count value for each file is stored and incremented at each time step. After each increment, the count value is compared to a master counter, the difference being the time delay in question.

Thus, if a file is active its incremented time
30 stamp is reset back to the initial value each time the data file is rewritten. In this manner, files that are constantly updated will have low time stamp identifiers and will be kept in cache until their activity
35 decreases. After a period of inactivity has expired,

0093133-091597

5

10

15

30

35

most active files (the top 5%), the write frequency of the Flash EEPROM may be reduced from the usual one every millisecond to one every 5 minutes. In this way the wear-out time for the memory can be extended almost indefinitely. This improvement is also accompanied by increased system performance during write.

Incorporating time tag into the write cache concept has the advantage that the size of the write cache buffer memory can be relatively small, since it is used only to store frequently written data files, with all other files written directly into the Flash EEPROM memory. A second advantage is that the management of moving data files in and out of the write cache buffer can be automated since it does not require advanced knowledge of which data files are to be called next.

The various aspects of the present invention that have been described co-operate in a system of Flash EEPROM memory array to make the Flash EEPROM memory a viable alternative to conventional non-volatile mass storage devices.

While the embodiments of the various aspects of the present invention that have been described are the preferred implementation, those skilled in the art will understand that variations thereof may also be possible. Therefore, the invention is entitled to protection within the full scope of the appended claims.

08031133 091597

IN THE CLAIMS:

1. A Flash EEprom system comprising:
one or more integrated circuit chips each
having an array of Flash EEprom cells partitioned into
a plurality of sectors, each sector addressable for
5 erase such that all cells therein are erasable
simultaneously;
means for selecting a plurality of sectors
among the one or more chips for erase operation; and
means for simultaneously performing the erase
10 operation on only the plurality of selected sectors.
2. A Flash EEprom system as in claim 1,
including
read or write operations on chips which have been
enabled by a chip select signal, wherein the erase
5 operation is performed on chips without regard to the
chip select signal.
3. A Flash EEprom system as in claim 1,
wherein the erase operation may be performed on the
plurality of sector selected for erase operation, while
read, write or other operations may be performed on any
5 other device not selected for erase operation.
4. The Flash EEprom system according to claim
1, further comprising:
means for individually removing any one or
combination of sectors from the plurality of selected
5 sectors, such that said removed sectors are prevented
from further erase during the erase operation.

2025-09-16 15:07:00

5. The Flash EEprom system according to claim 1, further comprising:

means for simultaneously deselecting all sectors.

6. The Flash EEprom system according to claim 1, wherein the selecting means further comprises:

individual register associated with each sector for holding a status to indicate whether the sector is
5 selected or not.

7. The Flash EEprom system according to claim 6, wherein the simultaneously erasing means is responsive to the status in each of the individual registers, such that only the selected sectors are
5 included in the erasing.

8. The Flash EEprom system according to claim 6, wherein any one or combination of the individual registers indicating a selected status are individually resettable to an un-selected status.

9. The Flash EEprom system according to claim 6, wherein all the individual registers are simultaneously resettable to a status indicating the associated sectors as not selected.

10. A system for correcting errors from defective cells within an array of Flash EEprom cells, comprising:

substitute cells;

5 means for substituting one or more of the defective cells with a corresponding number of substitute cells.

269760"EEFEE80

11. A system for correcting errors from defective cells within an array of Flash EEPROM cells as in claim 10 wherein the substituting means also applies automatically to new defective cells as soon as they are detected.

12. A system for correcting errors from defective cells within an array of Flash EEPROM cells as in claim 10, said array being partitioned into a plurality of Flash erasable sectors such that all cells within each sector are erasable at once, wherein the substitute cells are in the same sector as the defective cells.

13. A system for correcting errors from defective cells within an array of Flash EEPROM cells as in claim 11, further comprising a defect map for storing defect pointers which link the addresses of the defective cells to that of the corresponding substitute cells.

14. A system for correcting bad data in defective cells within an array of Flash EEPROM cells as in claim 13, wherein the defect map for said defective cells are located in the same sector as said defective cells.

2025-09-10 10:00:00

15. A system for correcting errors from defective cells within an array of Flash EEprom cells as in claim 10, said array being partitioned into a plurality of Flash erasable sectors such that all cells within each sector are erasable at once, wherein the substitute cells are in the same sector as the defective cells when the number of defective cells in the sector does not exceed a predetermined number, and the substitute cells are in a different sector when the number is exceeded.

16. A system for correcting errors from defective cells within an array of Flash EEprom cells as in claim 15, wherein said sector is replaced in its entirety by a substitute sector when said number is exceeded.

17. A system for correcting errors from defective cells within an array of Flash EEprom cells as in claim 15 wherein the substituting means also applies automatically new defective cells as soon as they are detected.

18. A system for correcting errors from defective cells within an array of Flash EEprom cells as in claim 17, including the use of error correction codes.

2025-09-17 10:00:00

19. A system for correcting bad data in defective cells within an array of Flash EEPROM cells, comprising:

substitute cells for storing good data intended
5 for the defective cells;

means for substituting the bad data in one or more of the defective cells with the good data in the corresponding substitute cells when the defective cells are accessed.

20. A system for correcting bad data in defective cells within an array of Flash EEPROM cells as in claim 19, further comprising means for automatically saving the good data intended to be written to the
5 defective cells to the corresponding substitute cells, thereby preserving the integrity of the good data.

21. A system for correcting bad data in defective cells within an array of Flash EEPROM cells as in claim 20 wherein the substituting means also applies automatically to new defective cells as soon as they are
5 detected.

22. A system for correcting bad data in defective cells within an array of Flash EEPROM cells as in claim 20, said array being partitioned into a plurality of Flash erasable sectors such that all cells
5 within each sector are erasable at once, and data is stored therein, wherein the substituting means applies after the data including the bad data has been accessed.

0893133, 091697
269160, 269160

23. A system for correcting bad data in defective cells within an array of Flash EEPROM cells as in claim 20, said array being partitioned into a plurality of Flash erasable sectors such that all cells within each sector are erasable at once, wherein the substitute cells are in the same sector as the defective cells.

24. A system for correcting bad data in defective cells within an array of Flash EEPROM cells as in claim 20, further comprising a defect map for storing defect pointers which link the addresses of the defective cells to that of the corresponding substitute cells.

25. A system for correcting bad data in defective cells within an array of Flash EEPROM cells as in claim 24, wherein the defect map for said defective cells are located in the same sector as said defective cells.

26. A system for correcting bad data in defective cells within an array of Flash EEPROM cells as in claim 19, said array being partitioned into a plurality of Flash erasable sectors such that all cells within each sector are erasable at once, wherein the substitute cells are in the same sector as the defective cells when the number of defective cells in the sector does not exceed a predetermined number, and the substitute cells are in a different sector when the number is exceeded.

269160-EE680

27. A system for correcting bad data in defective cells within an array of Flash EEPROM cells as in claim 26, wherein said sector is replaced in its entirety by a substitute sector when said number is exceeded.

28. A system for correcting bad data in defective cells within an array of Flash EEPROM cells as in claim 26, wherein the substituting means also applies automatically to newly occurring defective cells.

29. A system for correcting bad data in defective cells within an array of Flash EEPROM cells as in claim 28, including use of error correction codes.

30. An improved system for writing data files into a Flash EEPROM memory comprising:

a cache memory for temporarily storing data files intended for the Flash EEPROM memory, said cache memory able to undergo significantly more write/erase cycles than the Flash EEPROM memory;

means responsive to a system write to the Flash EEPROM memory for writing data files into the cache memory instead of the Flash EEPROM memory;

means for identifying each data file in the cache memory;

means for determining the time since each data file was last written; and

means for first moving data file having the longest time since last written from the cache memory to the Flash EEPROM memory when additional space for new data files is required in the cache memory, thereby substantially reducing the number of actual writes and associated stress to the Flash EEPROM memory.

0893133 "091697"

31. The improved system as in claim 30, further comprising:

a backup non-volatile memory for downloading the data files in the cache memory thereto; and

5 means responsive to an impending power loss for down loading the data files in the cache memory to the backup memory, thereby saving the data files from the possibly volatile cache memory.

32. The improved system as in claim 30, wherein the backup memory is part of the Flash EEPROM memory.

33. The improved system as in claim 30, wherein the cache memory has a significantly faster access time than that of the Flash EEPROM memory.

34. The improved system as in claim 30, including a controller circuit chip for controlling the operations of the Flash EEPROM memory, wherein the improved system is part of the controller circuit chip.

35. The improved system as in claim 30, including a microprocessor system and random access memory, wherein the improved system is implemented by software in the microprocessor system with random access
5 memory.

20091113, 091697

36. An improved system for writing data files into a Flash EEPROM memory comprising:

5 a cache memory for temporarily storing data files intended for the Flash EEPROM memory, said cache memory able to undergo significantly more write/erase cycles than the Flash EEPROM memory;

means responsive to a system write to the Flash EEPROM memory for writing data files into the cache memory instead of the Flash EEPROM memory;

10 a tag memory for storing the identity of data files and the time each data file was last written; and

means for first moving data file having the longest time since last written from the cache memory to the Flash EEPROM memory when additional space for new data files is required in the cache memory, thereby
15 substantially reducing the number of actual writes and associated stress to the Flash EEPROM memory.

37. The improved system as in claim 36, further comprising:

a backup non-volatile memory for downloading the data files in the cache memory thereto; and

5 means responsive to an impending power loss for down loading the data files in the cache memory to the backup memory, thereby saving the data files from the possibly volatile cache memory.

38. The improved system as in claim 36, wherein the backup memory is part of the Flash EEPROM memory.

39. The improved system as in claim 36, wherein the cache memory has a significantly faster access time than that of the Flash EEPROM memory.

08931133.091697

5

5

5

5

10

15

20

43. The improved system as in claim 42, further comprising:

a backup non-volatile memory for downloading the data files in the cache memory thereto; and

5 means responsive to an impending power loss for down loading the data files in the cache memory to the backup memory, thereby saving the data files from the possibly volatile cache memory.

44. The improved system as in claim 42, wherein the backup memory is part of the Flash EEPROM memory.

45. The improved system for writing data files into a Flash EEPROM memory according to claim 42, wherein said responsive means for writing includes a tag memory for storing the identity of data files and the
5 time each data file was last written, and wherein said responsive means writing to the Flash EEPROM when said data file is not tagged in the tag memory, and writing to the cache memory when said data file is tagged in the tag memory.

009343 094597
469460" EET E680

46. An improved system for writing data files into a Flash EEPROM memory comprising:

5 a cache memory for temporarily storing data files intended for the Flash EEPROM memory, said cache memory able to undergo significantly more write/erase cycles than the Flash EEPROM memory;

10 means responsive to a system write to the Flash EEPROM memory for writing a data file either into the Flash EEPROM memory or instead into the cache memory, said responsive means writing to the Flash EEPROM when said data file is last written after the predetermined period of time, and writing to the cache memory when said data file is last written within a predetermined period of time; and

15 means for first moving data files having the longest times since last written from the cache memory to the Flash EEPROM memory when additional space for new data files is required in the cache memory, thereby substantially reducing the number of actual writes and
20 associated stress to the Flash EEPROM memory.

47. The improved system as in claim 46, wherein the cache memory has a significantly faster access time than that of the Flash EEPROM memory.

48. The improved system as in claim 46, including a controller circuit chip for controlling the operations of the Flash EEPROM memory, wherein the improved system is part of the controller circuit chip.

49. The improved system as in claim 46, including a microprocessor system and random access memory, wherein the improved system is implemented by software in the microprocessor system with random access
5 memory.

50. An improved system for writing data files into a Flash EEPROM memory comprising:

a cache memory for temporarily storing data files intended for the Flash EEPROM memory, said cache
5 memory able to undergo significantly more write/erase cycles than the Flash EEPROM memory;

a tag memory for storing the identity of data files and the time each data file was last written;

means responsive to a system write to the Flash
10 EEPROM memory for writing a data file either into the Flash EEPROM memory or instead into the cache memory, said responsive means writing to the Flash EEPROM when the data file is not identified in the tag memory, and writing to the cache memory when the data file is
15 identified in the tag memory; and

means for moving first the data files having the longest times since last written from the cache memory to the Flash EEPROM memory when additional space for new data files is required in the cache memory,
20 thereby substantially reducing the number of actual writes and associated stress to the Flash EEPROM memory.

51. The improved system as in claim 50, further comprising:

a backup non-volatile memory for downloading the data files in the cache memory thereto; and

5 means responsive to an impending power loss for down loading the data files in the cache memory to the backup memory, thereby saving the data files from the possibly volatile cache memory.

52. The improved system as in claim 50, wherein the backup memory is part of the Flash EEPROM memory.

091133 091657

53. The improved system as in claim 50, wherein the cache memory has a significantly faster access time than that of the Flash EEPROM memory.

54. The improved system as in claim 50, including a controller circuit chip for controlling the operations of the Flash EEPROM memory, wherein the improved system is part of the controller circuit chip.

55. The improved system as in claim 50, including a microprocessor system and random access memory, wherein the improved system is implemented by software in the microprocessor system with random access memory.

56. A memory card adapted to plug into a computer system in a manner to communicate with a system bus and a standard power supply, comprising the following mounted thereon:

5 a plurality of EEprom integrated circuit chips, each of said chips including:

10 a large number of individually addressable storage cells organized into a plurality of sectors, each sector containing a plurality of said storage cells,

a plurality of spare storage cells within any of said sectors,

15 means responsive to signals on said system bus for erasing all cells in one or more designated sectors without erasing cells in others of said sectors,

means responsive to signals on said system bus for reading the state of addressed storage cells,

20 means responsive to signals on said system bus for programming addressed storage cells to a predetermined state, and

25 means responsive to an unsuccessful attempt to either program or erase a storage cell within one of said sectors for substituting one of said spare storage cells therefore while maintaining operation of the remaining cells of said sector.

08931133.091597

57. The memory card according to claim 56 which additionally comprises a cache memory mounted on said card, and wherein said programming means includes means for initially programming said cache memory rather than said EEprom memory, said reading means includes means for initially determining whether the cache memory contains data to be read, and said programming means additionally includes means responsive to said cache memory becoming full for writing its oldest unused block of data into said EEprom memory, thereby to make room for new data in said cache memory.

58. The memory card as in claim 56, wherein each of said chips further includes a plurality of spare sectors, and wherein said substituting means also substitutes one of said spare sectors for one of said sectors when a predetermined number of cells in said one of said sector become defective.

59. The memory card as in claim 58, including means for performing error correction using error correction codes.

60. The memory card as in claim 56, including a controller and an interface connected to the system bus, said controller being adapted to be responsive to commands intended for a standard magnetic disk drive storage system connectable to the computer system, thereby emulating said disk drive system.

2025 RELEASE UNDER E.O. 14176

61. The memory card as in claim 56, in which various operating voltages are required for various operations of the EEprom chips, including means for generating the various operating voltages from the standard power supply.

62. A storage system incorporating therein the memory card of claim 56, comprising:

a controller for controlling the operation of the EEprom chips;

means for generating voltages for the operation of the EEprom chips;

means for error correction in the operation of the storage system; and

means for interfacing the storage system to a computer system.

08934433-094697

ABSTRACT

A system of Flash EEprom memory chips with controlling circuits serves as non-volatile memory such as that provided by magnetic disk drives. Improvements include selective multiple sector erase, in which any combinations of Flash sectors may be erased together. Selective sectors among the selected combination may also be de-selected during the erase operation. Another improvement is the ability to remap and replace defective cells with substitute cells. The remapping is performed automatically as soon as a defective cell is detected. When the number of defects in a Flash sector becomes large, the whole sector is remapped. Yet another improvement is the use of a write cache to reduce the number of writes to the Flash EEprom memory, thereby minimizing the stress to the device from undergoing too many write/erase cycling.

08931133-091697

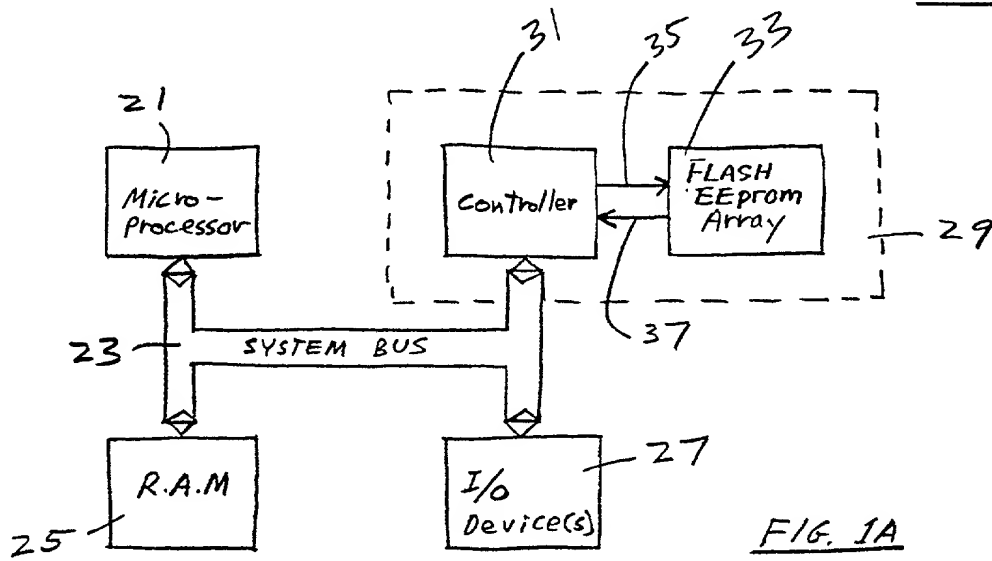


FIG. 1A

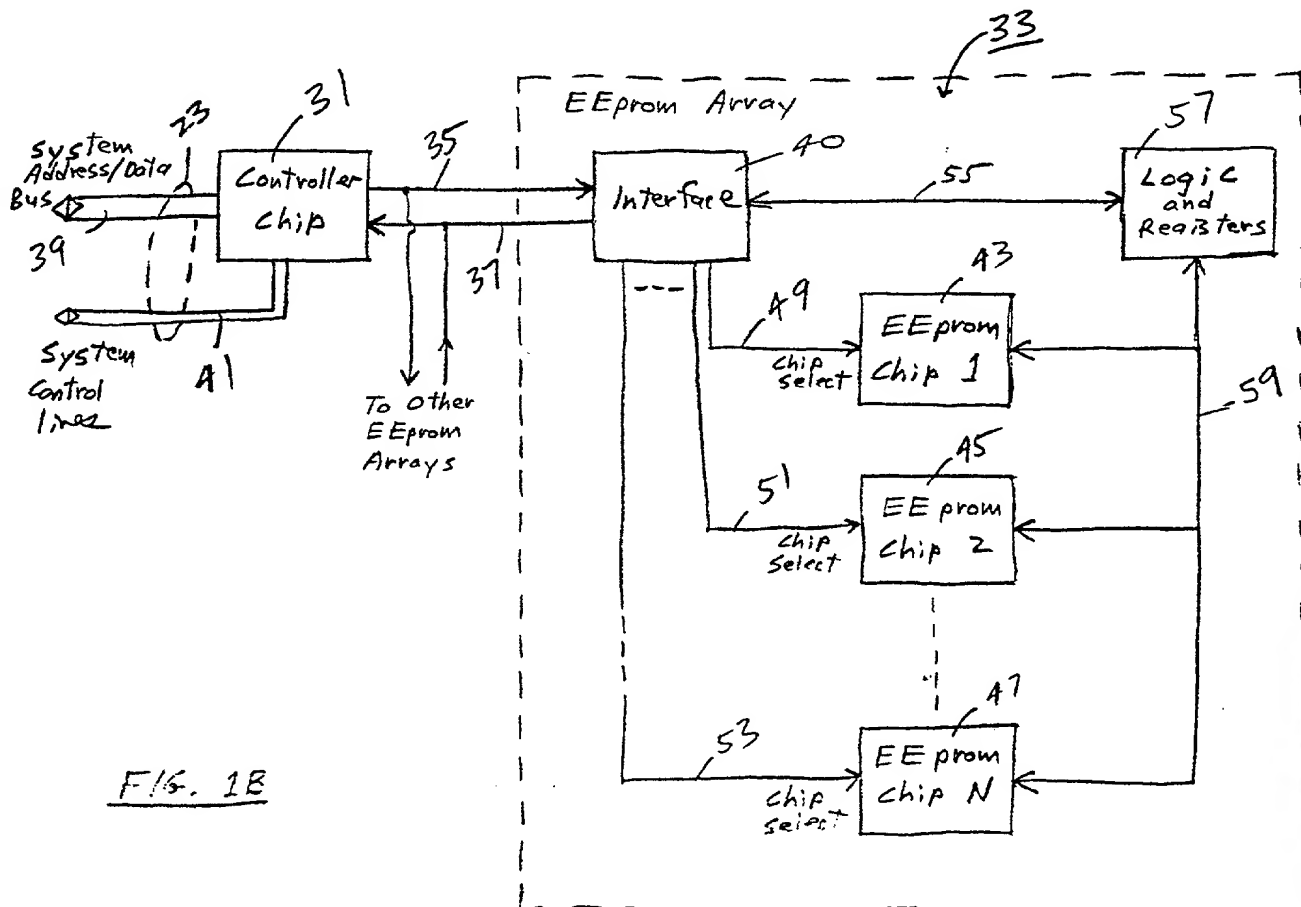


FIG. 1B

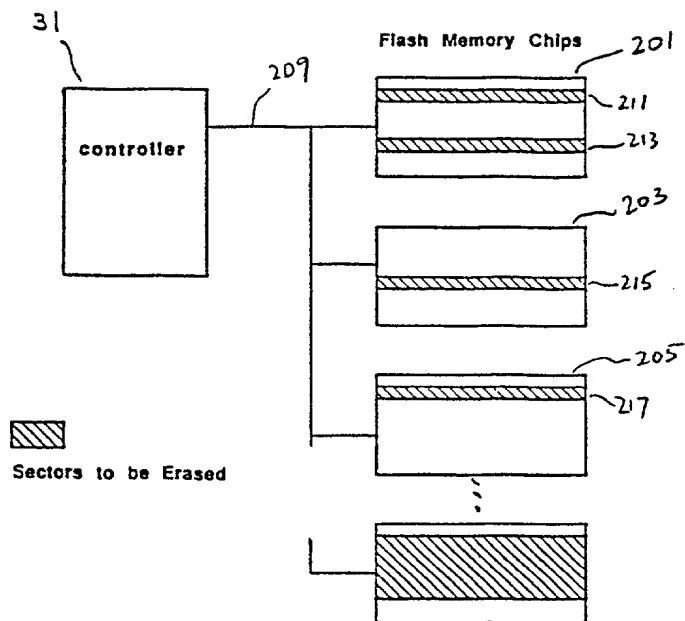


FIG-2

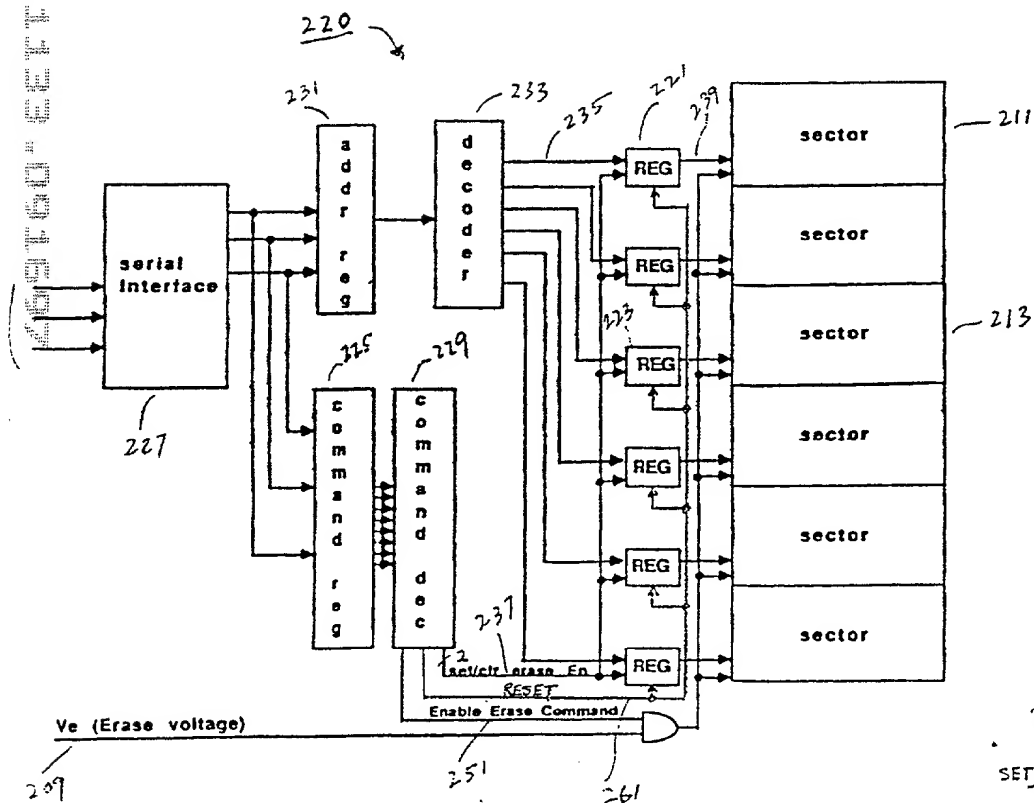


FIG-3A

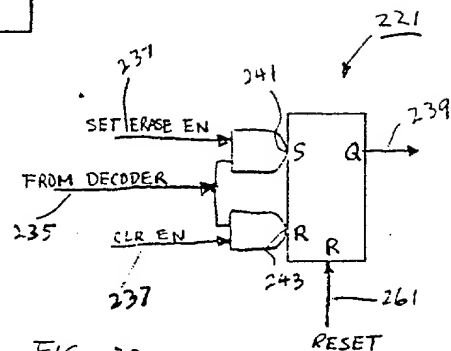


FIG-3B

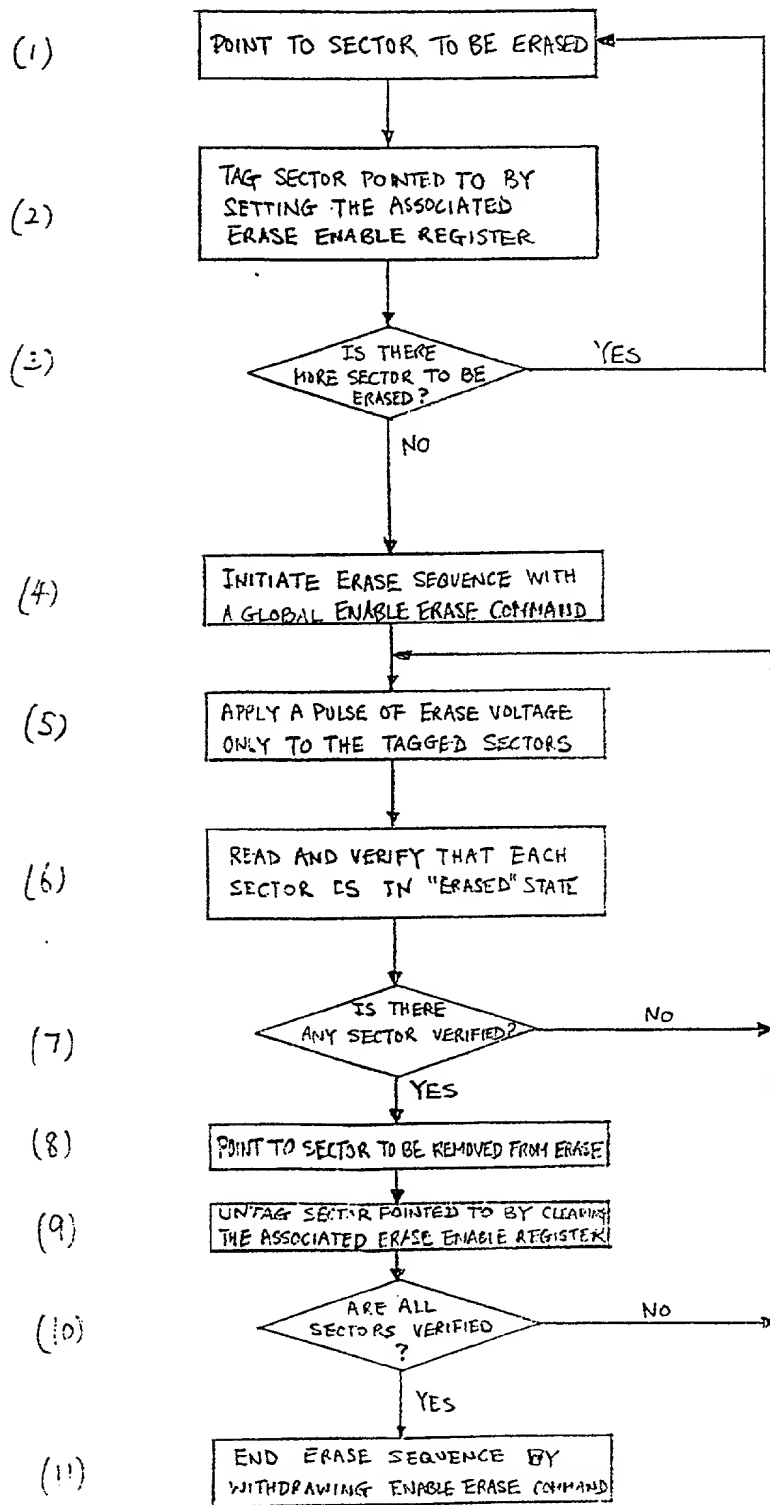
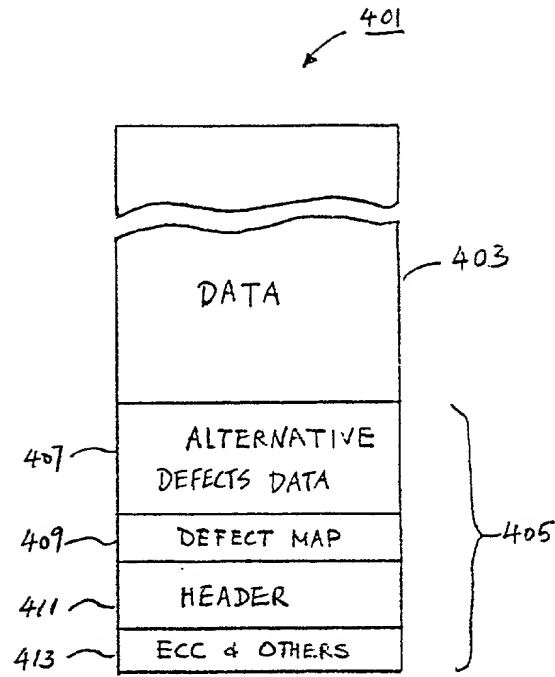


FIG. 4



SECTOR PARTITION

FIG-5

0831133-091697

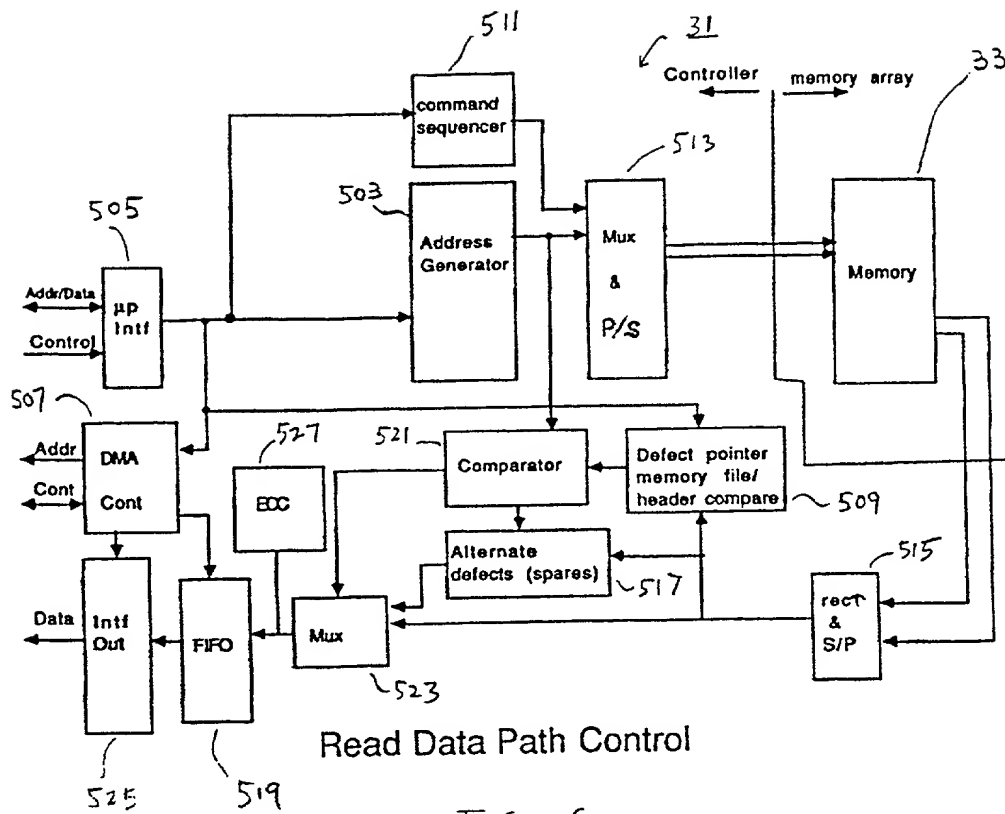


FIG-6

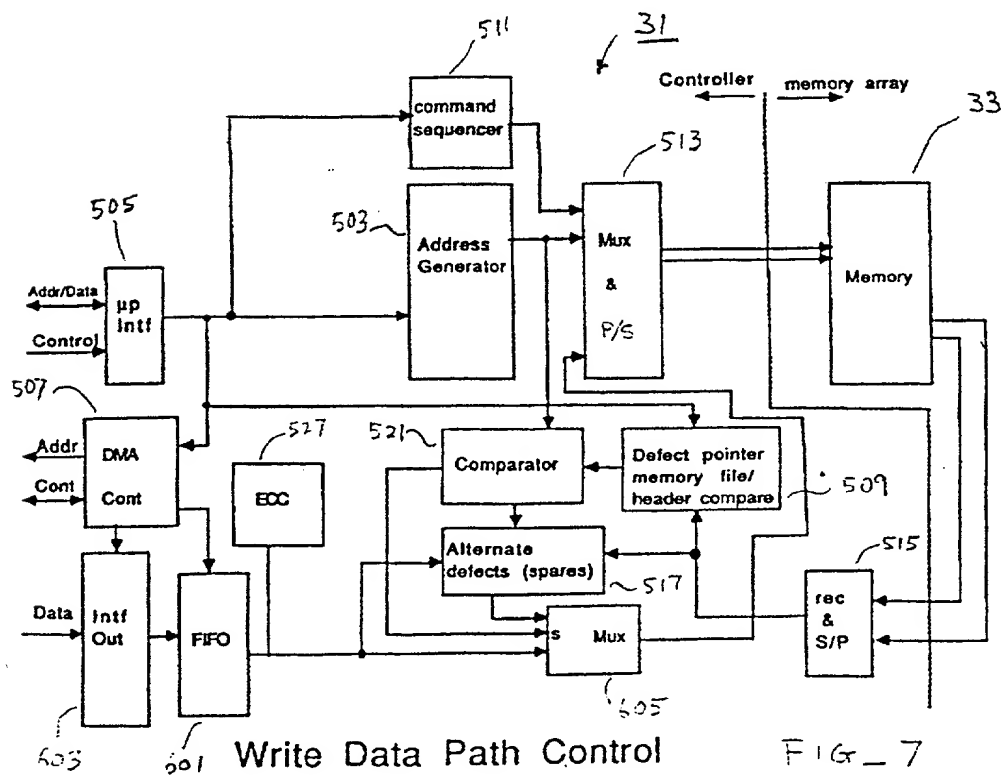


FIG-7

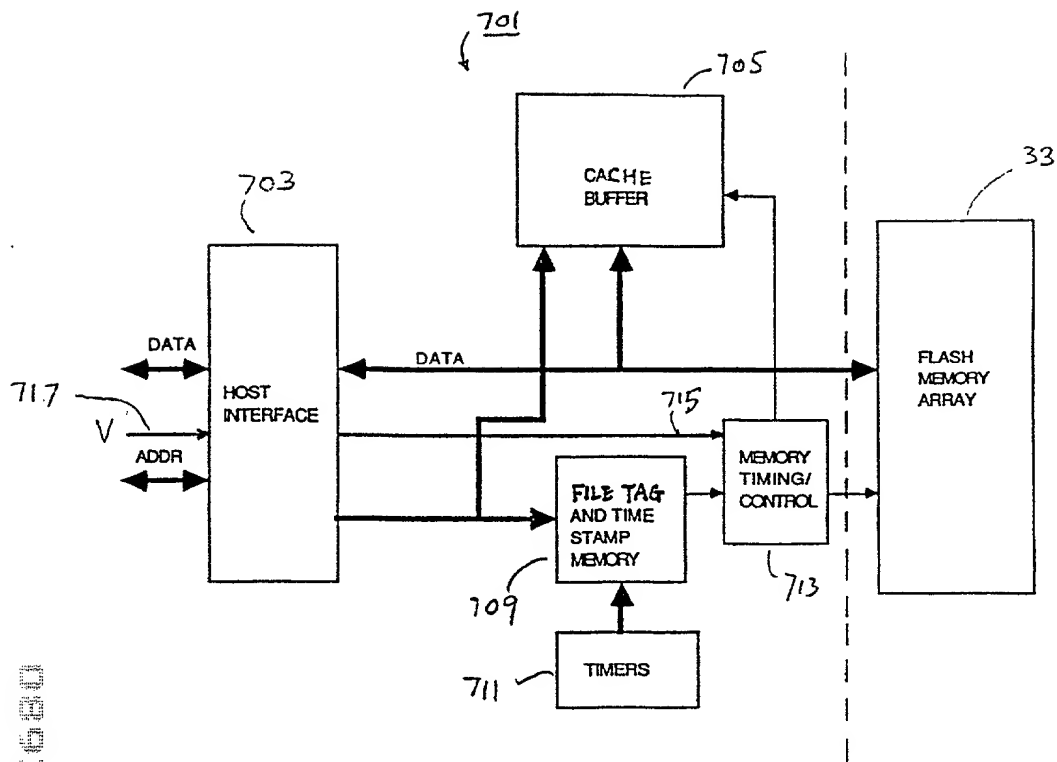


FIG. 8

PATENT APPLICATION DECLARATION
(Attorney's Docket No.: HARI-0600)

Each of the Applicants named below hereby declares as follows:

1. My residence, post office address and country of citizenship given below are true and correct.

2. I believe I am the original, first and joint inventor of the invention claimed in the patent application specification Serial No. 337,566, filed April 13, 1989, for which a patent is sought, and I have reviewed and understand the contents of the attached specification, including its claims.

3. I acknowledge my duty to disclose information of which I am aware which is material to the examination of this application. I understand that information is material where there is a substantial likelihood that a reasonable patent examiner would consider it important in deciding whether to allow the attached application to issue as a patent.

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Additionally, the undersigned hereby appoint the following as attorneys and agents to prosecute said patent application, to transact all business in the Patent and Trademark Office connected therewith, to receive the original Letters Patent and to substitute or associate other attorneys on my behalf:

Gerald P. Parsons
Martin F. Majestic
J. Suzanne Siebert
James S. Hsue
Philip Yau

Registration No. 24,486
Registration No. 25,695
Registration No. 28,758
Registration No. 29,545
Registration No. 32,892

Please send all correspondence to:

Philip Yau
MAJESTIC, PARSONS, SIEBERT & HSUE
Four Embarcadero Center, Suite 1450
San Francisco, CA 94111-4121

Telephone: 415/362-5556
Facsimile: 415/362-5418

Date: 5.15, 1989 Eliyahon Harari
Eliyahon Harari

Residence and
Post Office Address: 104 Auzerai Court
Los Gatos, CA 95030
(Citizenship: Israel)

Date: 5.16, 1989 Robert D Norman
Robert D. Norman

Residence and
Post Office Address: 6656 Pebblewood Court
San Jose, CA 95120
(Citizenship: United States)

Date: 5.15., 1989 Sanjay Mehrotra
Sanjay Mehrotra

Residence and
Post Office Address: 735 Berkshire Place
Milpitas, CA 95035
(Citizenship: India)

0591133 091697

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of)	
)	
ELIYAHOU HARARI, ROBERT D.)	Parent Application
NORMAN, and SANJAY MEHROTRA)	Assigned to Group Art
)	Unit: 2413
Serial No.: UNASSIGNED)	
)	
Filed: HEREWITH)	
)	
For: FLASH EEPROM SYSTEM)	
_____)	San Francisco, California

Hon. Commissioner of
Patents and Trademarks
Washington, D.C. 20231


TRANSMITTAL OF FORMAL DRAWINGS

Sir:

Please substitute the attached formal drawings, comprising five (5) sheets, for the informal drawings presently of record.

Respectfully submitted,

Dated: September 16, 1997



Gerald P. Parsons, Reg. No. 24,486
MAJESTIC, PARSONS, SIEBERT & HSUE
Four Embarcadero Center, Suite 1100
San Francisco, CA 94111-4106
Telephone: (415) 248-5500
Facsimile: (415) 362-5418

Atty. Docket: HARI.006USG

009344060-00000000

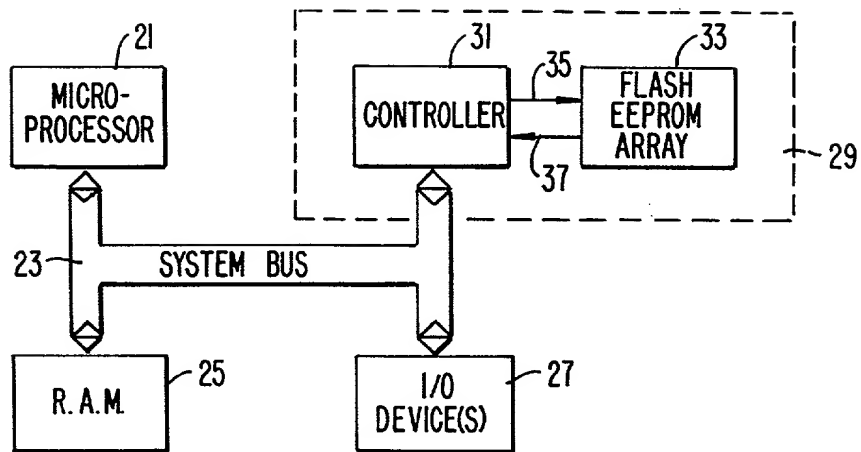


FIG. 1A.

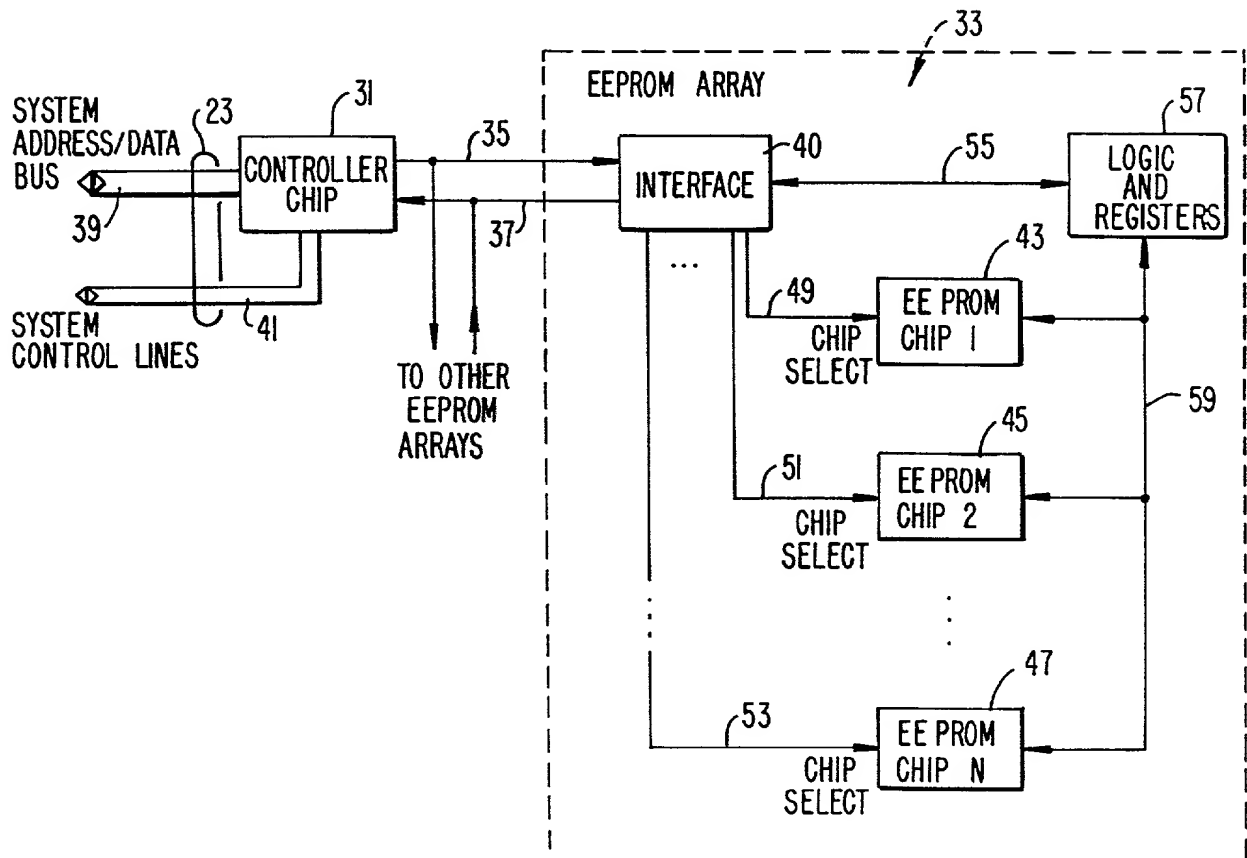


FIG. 1B.

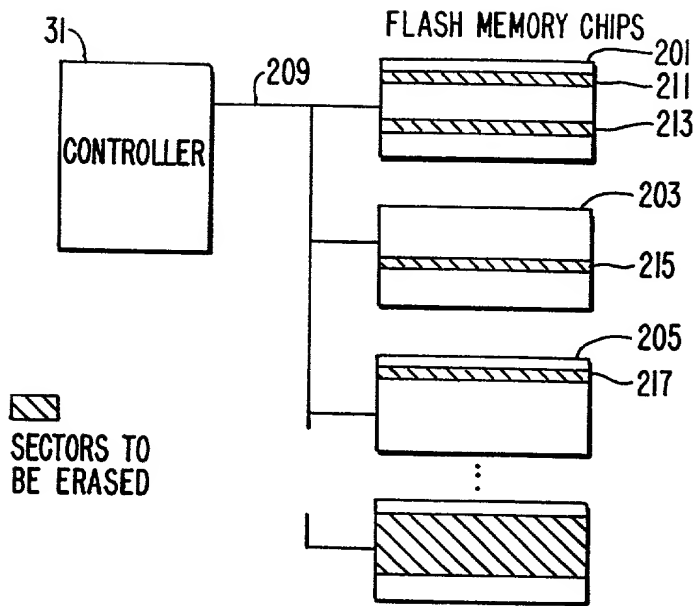


FIG. 2.

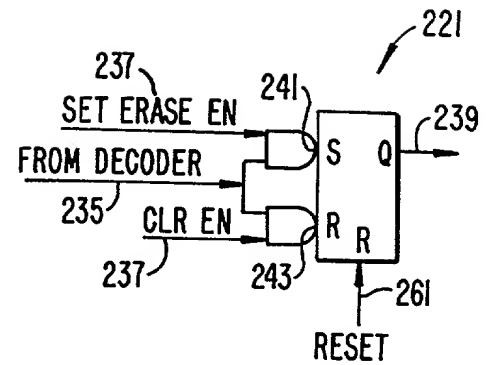


FIG. 3B.

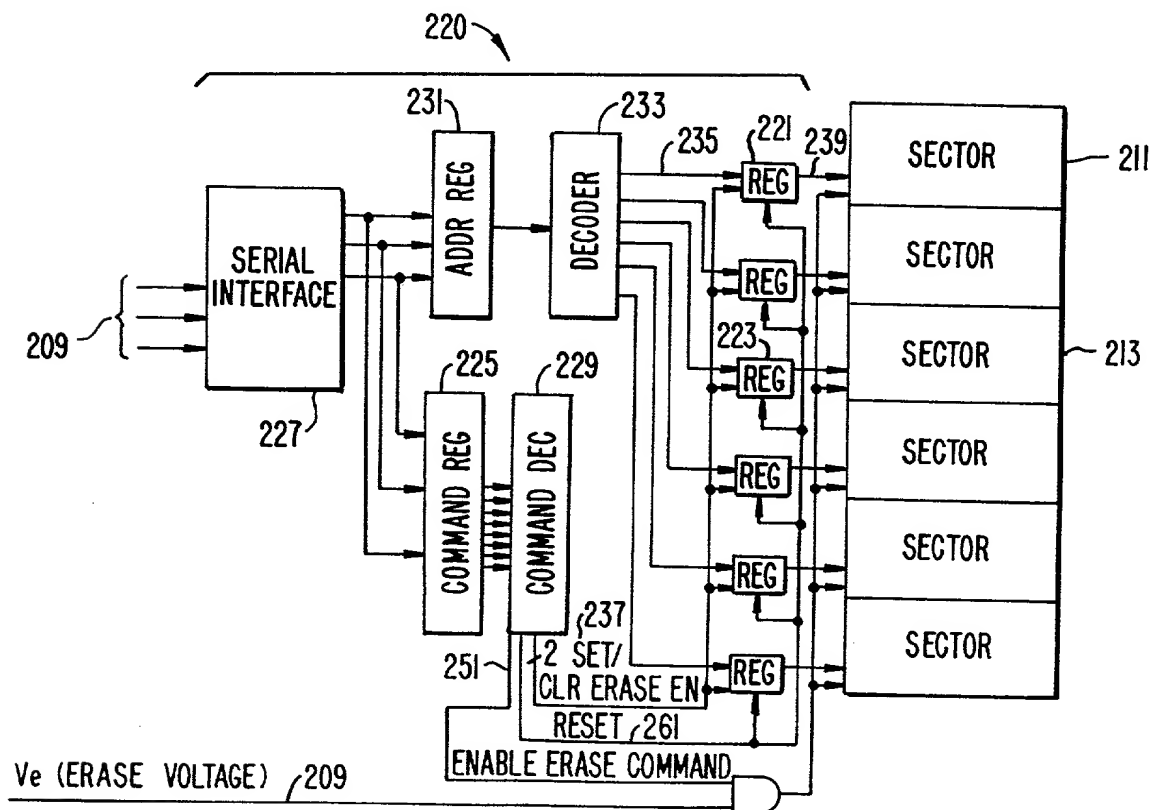


FIG. 3A.

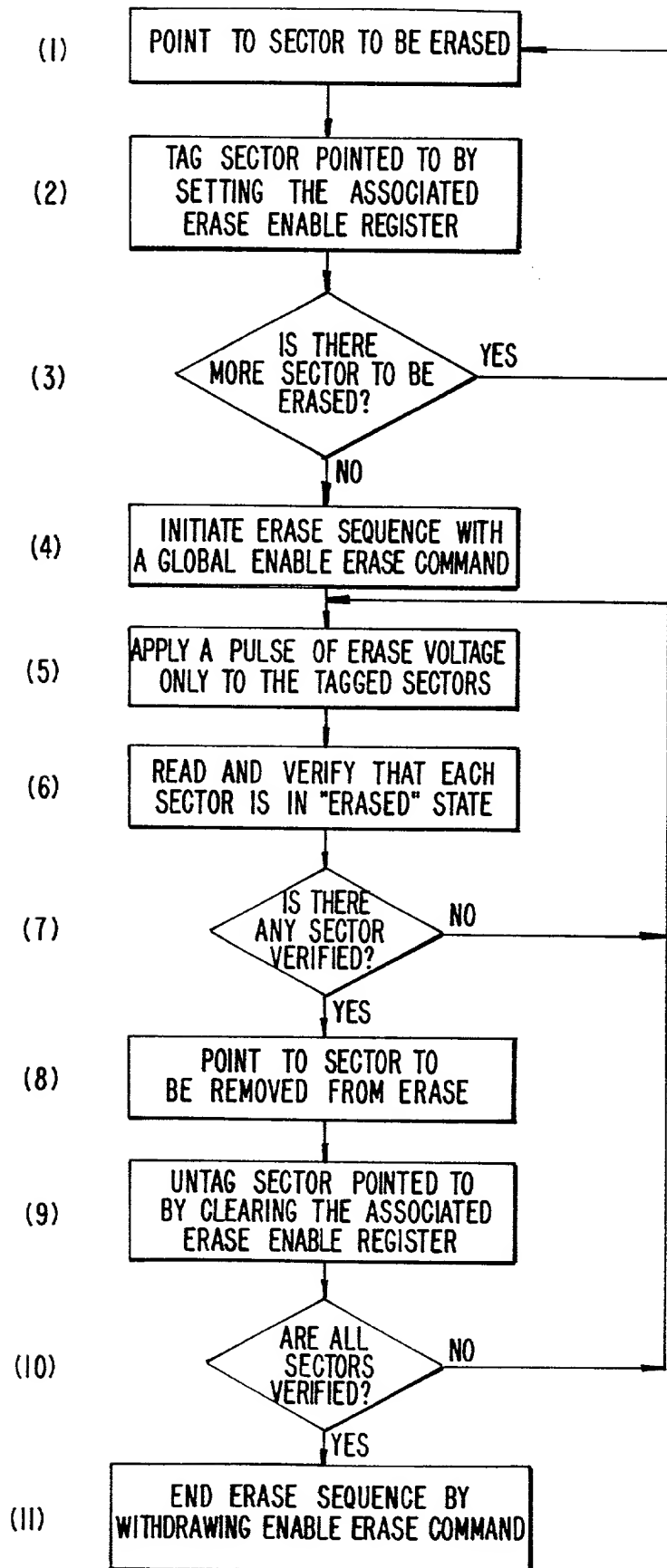
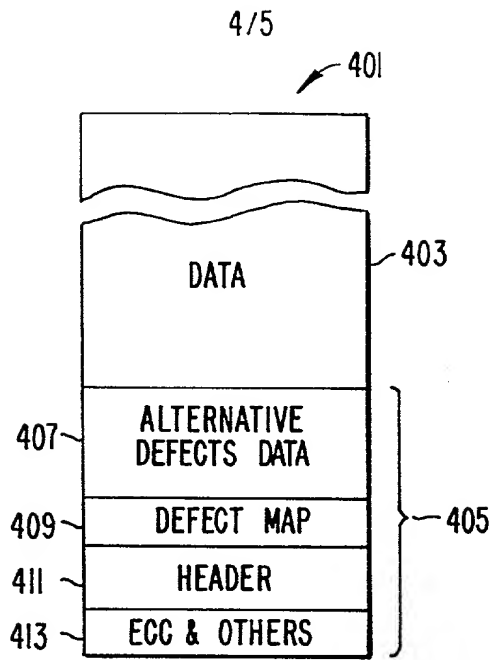


FIG._4.



SECTOR PARTITION

FIG. 5.

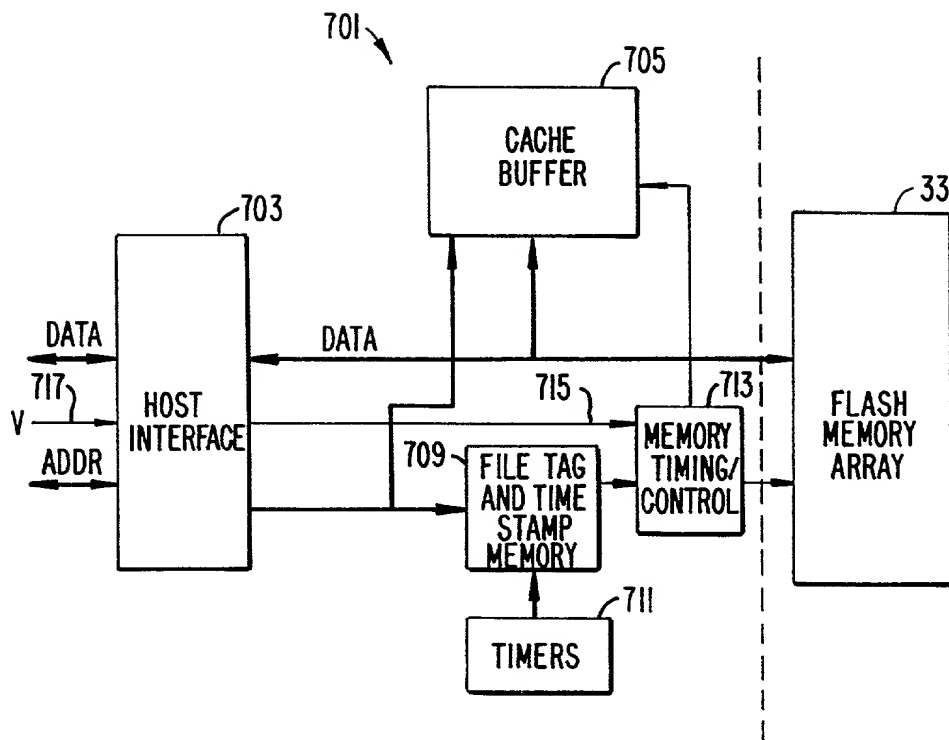
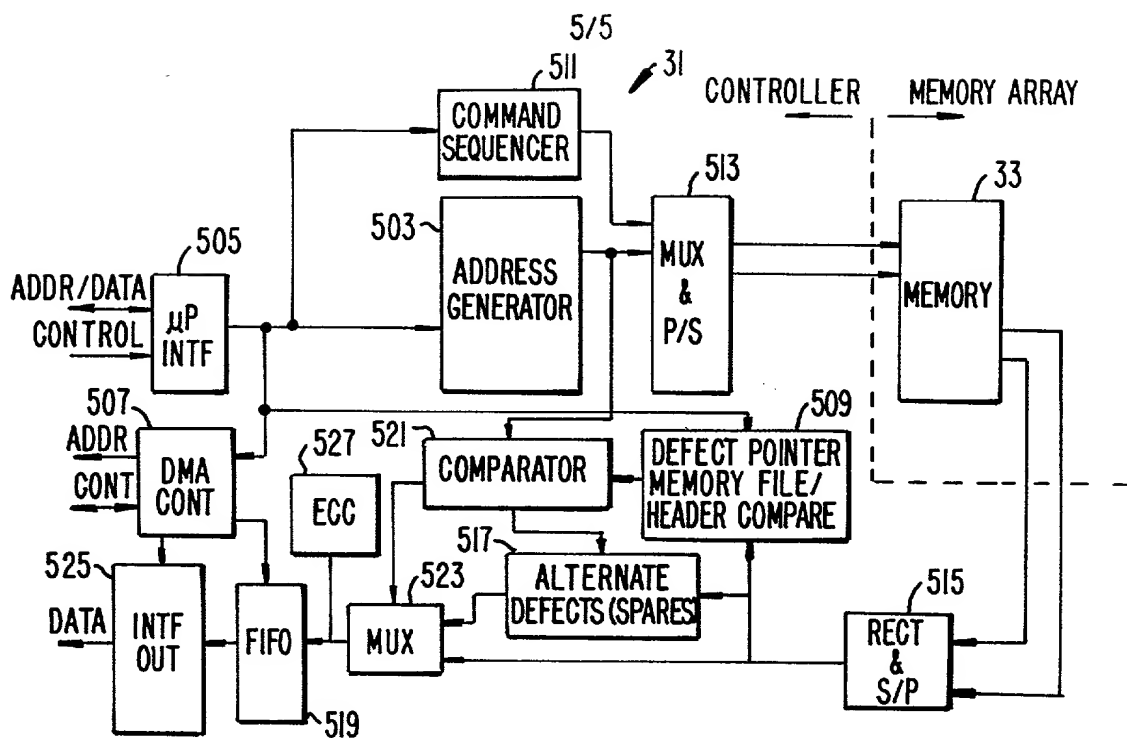
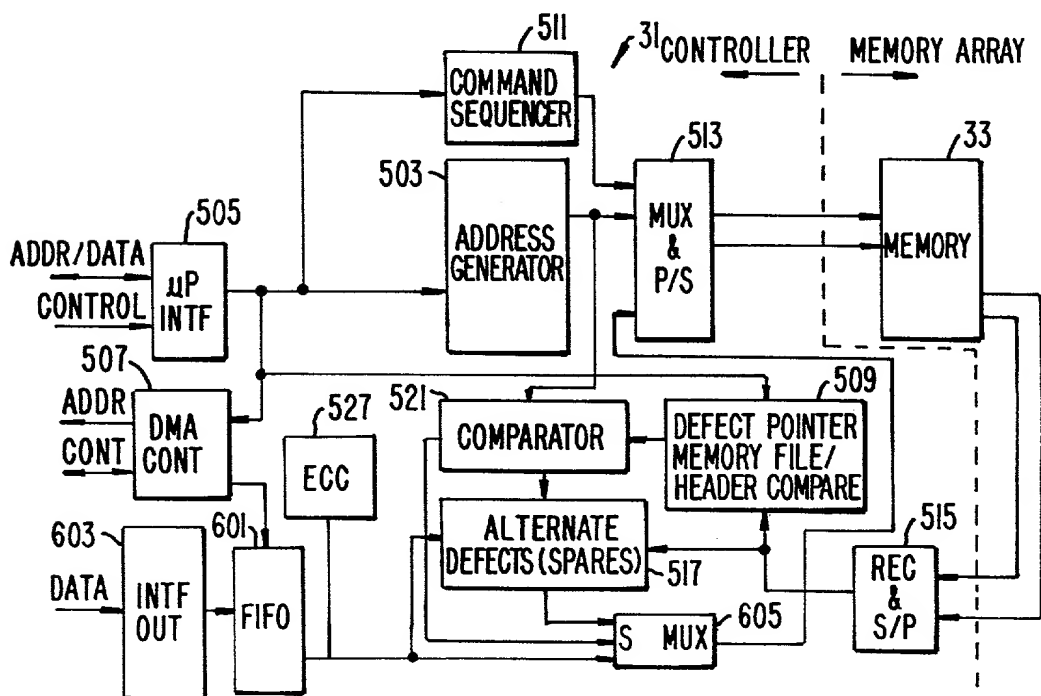


FIG. 8.



READ DATA PATH CONTROL

FIG. 6.



WRITE DATA PATH CONTROL

FIG. 7.